Managing data through the lens of an ontology

Maurizio Lenzerini
Dipartimento di Ingegneria Informatica,
Automatica e Gestionale Antonio Ruberti
Università di Roma La Sapienza
lenzerini@diag.uniroma1.it

Abstract

Ontology-based data management aims at managing data through the lens of an ontology, i.e., a conceptual representation of the domain of interest in the underlying information system. This new paradigm provides several interesting features, many of which have been already proved effective in managing complex information systems. This paper provides an introduction to the notion ontology-based data management, illustrating the main ideas underlying the paradigm, and pointing out the importance of the two areas of Knowledge Representation and Automated Reasoning for addressing the technical challenges it introduces.

Introduction

While the amount of data stored in current information systems continuously grows, and the processes making use of such data become more and more complex, extracting knowledge and getting insights from these data, as well as governing both data and the associated processes, are still challenging tasks. The problem is complicated by the proliferation of data sources and services both within a single organization, and in cooperating environments. Moreover, if we add to the picture the (inevitable) need of dealing with big data, and consider in particular the two Vs of "volume" and "velocity", we can easily understand why effectively accessing, integrating and managing data in complex organizations is still one of the main issues faced by Information Technology (IT) industry nowadays. Indeed, it is not surprising that data scientists spend a comparatively large amount of time in the data preparation phase of a project, compared with the data minining and knowledge discovery phase. Whether you call it data wrangling, data munging, or data integration, it is estimated that 50%-80% of a data scientists time is spent on collecting and organizing data for analysis[1]. If we consider that in any complex organization, data governance is also essential for tasks other than data analytics, we can conclude that the challenge of identifying, gathering, retaining, and providing access to all relevant data for the business at an acceptable cost, is huge (Bernstein and Haas 2008).

[1]The 2017 Data Scientist Report, CrowdFlower.

The above considerations are valid even for very simple information systems, as the following example scenario illustrates. Figure 1 shows a portion of `Cust_table`, a relational table contained in a real information system. The table maintains information about the customers of an organization, where each row stores data about a single customer. The first column contains the code of the customer, with the proviso that if the code is positive, then the record refers to an ordinary customer, and if it is negative, to a special customer. If the code is non-numeric, then the type of the customer is unknown. Columns 2 and 3 specify the time interval of validity for the record, `ID_GROUP` indicates the group the customer belongs to (if the value of `FLAG_CP` is "S", then the customer is the leader of the group, and if `FLAG_CF` is "S", then the customer is the controller of the group), `FATTURATO` is the annual turnover (but the value is valid only if `FLAG_FATT` is "S"). Obviously, each notion mentioned above (like "special", "ordinary" "group", "leader", etc.) has a specific meaning in the organization, and understanding such meaning is crucial if one wants to correctly access or manage the data in the table and extract useful information out of it. Similar rules hold for the other 47 columns that, for lack of space, are not shown in the figure.

Those who have experience of complex databases, or databases that are part of large information systems will not be surprised to see such complexity in a single data structure. Now, think of a database with many tables of this kind, and try to imagine a poor client accessing such tables for data analysis. The problem is even more severe if one con-

CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S
-472900	10-mag-2001	1-gen-9999	62010	S	N	0,00	N
130976	7-mag-2001	9-lug-2003	75680				

Figure 1: Fragment of the `Cust_table` table

siders that information systems in the real world use different (often many) heterogeneous data sources, both internal and external to the organization. While many are the issues raised by this problem, I would like to go in more detail on some of them.

Accessing and querying data. As observed in (De Giacomo et al. 2018), although the initial design of a collection of data sources might be adequate, corrective maintenance actions tend to re-shape them into a form that often diverges from the original structure. Also, they are often subject to changes so as to adapt to specific, application-dependent needs. Analogously, applications are continuously modified for accommodating new requirements, and guaranteeing their seamless usage within the organization is costly. The result is that the data stored in different sources and the processes operating over them tend to be redundant, mutually inconsistent, and obscure for large classes of users. So, query formulation often requires interacting with IT experts who knows where the data are and what they mean in the various contexts, and can therefore translate the information need expressed by the user into appropriate queries. It is not rare to see organizations where this processes requires domain experts to send a request to the data management staff and wait for several days (or even weeks) before they receive a (possibly inappropriate) query in response. In summary, it is often exceedingly difficult for end users to single out exactly the data that are relevant for them, even though they are perfectly able to describe their requirement in terms of business concepts.

Data quality. It is often claimed that data quality is one of the most important factors in delivering high value information services (Fan and Geerts 2012). However, the abovementioned scenario poses several obstacles to the goal of even checking data quality, let alone achieving a good level of quality in information delivery. How can we possibly specify data quality requirements, if we do not have a clear understanding of the semantics that data should bring? The problem is sharpened by the need of connecting to external data, originating, for example, from business partners, suppliers, clients, or even public sources. Again, judging about the quality of external data, and deciding whether to reconcile possible inconsistencies or simply adding such data as different views, cannot be done without a deep understanding of their meaning.

Open data. Note that understanding and documenting the semantics of data is also crucial for opening data to external organizations. The demand of greater openness is irresistible nowadays. In many aspects of our society there is growing awareness and consent on the need for data-driven approaches that are resilient, transparent and fully accountable. But to achieve a data-driven society, it is necessary that the data needed for public goods are readily available (Wessels et al. 2017). Thus, it is no surprising that in recent years, both public and private organizations have been faced with the issue of publishing Open Data, in particular with the goal of providing data consumers with suitable information to capture the semantics of the data they publish. But again, associating a reasonably well-structured description of open datasets is very hard if we do not have effective tools for documenting the meaning and the usage of the data sources from which such data have been extracted.

Process and service specification. Information systems are crucial artifacts for running organizations, and designing, documenting, managing, and executing processes is an important aspect of information systems. However, specifying what a process/service does, or which characteristics it is supposed to have, cannot be done correctly and comprehensively without a clear specification of which data the process will access, and how it will possibly modify or update such data. The difficulties of doing that in a satisfactory way come from various factors, including the lack of modeling languages and tools for describing process and data holistically. However, the problems related to the semantics of data that we discussed above undoubtedly makes the task even harder (Berardi et al. 2003; Bagheri Hariri et al. 2013).

The notion of Ontology-based Data Management system

All the above observations show that a unified access to data, a comprehensive methodology for data preparation, and an effective governance of data-oriented processes and services are extremely difficult goals to achieve in modern information systems (Bernstein and Haas 2008). We argue that the ontology-based data management (OBDM[2]) paradigm (Lenzerini 2011) is a promising direction for addressing the above challenges. The key idea of OBDM is to apply suitable techniques from the area of Knowledge Representation and Reasoning in Artificial Intelligence for a new way to achieve data governance and integration, based on the principle of managing heterogeneous data through the lens of an ontology. Indeed, OBDM resorts to a three-level architecture, constituted by the ontology, the data sources, and the mapping between the two:

- The *data layer* is constituted by the existing data sources that are relevant for the organization.
- The *ontology* is a declarative and explicit representation of the domain of interest for the organization, specified by means of a formal and high level description of both its static and dynamic aspects.
- The *mapping* is a set of declarative assertions specifying how the available sources in the data layer and the computational resources used in the organization relate to the ontology.

OBDM can thus be seen as a sophisticated form of information integration (Lenzerini 2002; Calvanese and De Giacomo 2005; Doan, Halevy, and Ives 2012), where the usual global schema is replaced by the conceptual model of the application domain, formulated as an ontology. With this approach, the integrated view that the system provides

[2]The acronym is similar to OBDA, which stands for Ontology-based Data Acces. We use OBDM, because we consider data access to be just one aspect, although important, of the more general notion of data management.

to information consumers is not merely a data structure accommodating the various data at the sources, but a semantically rich description of the relevant concepts in the domain of interest, as well as the relationships between such concepts. The distinguishing feature of the whole approach is that users of the system will be freed from all the details of how to use the data sources, as they will express their needs (e.g., a query) in the terms of the concepts, the relations, and the processes described in the domain model. The system will reason about the ontology and the mappings, and will reformulate the needs in terms of appropriate calls to services provided for accessing the data sources. In order to translate the services expressed over the ontology into correct and efficient computations over the data sources, techniques typical of the two areas of Knowledge Representation and Automated Reasoning are crucial. Note, however, that OBDM introduces new challenges to these areas. Indeed, while Knowledge Representation techniques are often confined to scenarios where the complexity resides in the rules governing the application, in OBDM one faces the problem of a huge amount of data in the data layer, and this poses completely new requirements for the reasoning tasks that the system should be able to carry out. For example, the notion of data complexity, by which one measures the computational complexity on the basis of the size of the data layer only, is of paramount importance in OBDM.

From a more formal perspective, an *OBDM specification* $\mathcal{J}$ is defined as a triple $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ is an ontology, $\mathcal{S}$ is a relational schema, called source schema, and $\mathcal{M}$ is a mapping from $\mathcal{S}$ to $\mathcal{O}$. In particular, $\mathcal{O}$ represents intensional knowledge about the domain, expressed in some logical language[3], and $\mathcal{M}$ is a set of mapping assertions, again expressed in a logical language, each one relating a query over the source schema to a query over the ontology.

An *OBDM system* is a pair $(\mathcal{J}, D)$ where $\mathcal{J}$ is an OBDM specification and D is a database for the source schema $\mathcal{S}$, called source database for $\mathcal{J}$. The semantics of $(\mathcal{J}, D)$ is given in terms of the logical interpretations that are models of $\mathcal{O}$, i.e., that satisfy all axioms of $\mathcal{O}$ and all assertions in $\mathcal{M}$ with respect to D. The notion of mapping satisfaction depends on the semantic interpretation adopted for mapping assertions. Commonly, such assertions are assumed to be *sound*, which intuitively means that the patterns specified over the sources imply a set of facts at the ontology level; in other words, data at the sources give rise to instance assertions in the ontology. Because of the logical nature of the domain description represented by the ontology, and the kind of mapping assertions considered, $(\mathcal{J}, D)$ is characterized by a set of models, denoted with $Mod_D(\mathcal{J})$.

We end this section by illustrating a simple example of an OBDM specification, referring in particular to the application scenario mentioned in section 1. We will use the example in the next sections.

- Source schema $\mathcal{S}$: we assume that, beside the table `Cust_table` illustrated in figure 1, we have another relational table available, called `Inv_table`, that stores pairs $\langle$`C1`,`C2`$\rangle$ such that the customer with code `C1` involved customer with code `C2` in a joint project. So, $\mathcal{S}$ is constitued by the relational schema $\{$`Cust_table`, `Inv_table`$\}$.
- Ontology $\mathcal{O}$: a fragment of the domain ontology expressed in graphical form is shown in figure 2. The ontology sanctions that there are exactly two types of customers, namely ordinary and special, so that every customer is of one of these types. Also, the ontology defines `Involved` as a relationship between customers.
- Mapping $\mathcal{M}$: the mapping, also shown in figure 2, asserts that the `Cust_table` table is mapped to the concepts `Ordinary` and `Special`, depending on the value of the field `CUC`, while data in the `Inv_table` are mapped to the relation `Involved`.

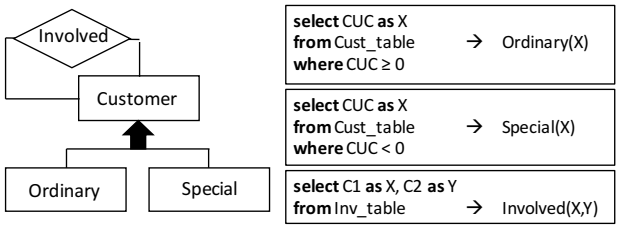

Figure 2: Example of OBDM specification: ontology and mapping

We obtain an OBDM system by pairing the above specification with a specific $\mathcal{S}$-database, i.e., a database coherent with the schemas $\mathcal{S}$, that assigns an extension (set of tuples) to the tables `Cust_table` and `Inv_table`.

Query answering

In OBDM systems, the main service of interest is *query answering*, i.e., computing the answers to user queries, which are queries posed over the ontology. It amounts to return the so-called *certain answers*, i.e., the tuples that satisfy the user query in all the models in $Mod_D(\mathcal{J})$. Notice the difference with query answering in traditional databases. While a database can be seen as a single model of a logical theory (see (Reiter 1984)), query answering in OBDM faces the problem of considering various models of the whole system, and is therefore a form of reasoning under incomplete information. It follows that it is much more challenging than classical query evaluation over a database instance, and this explains why automated deduction techniques are very relevant in this context.

To better illustrate the point, we reconsider the example of the previous section, and we assume that in a specific $\mathcal{S}$-database, `-452901` and `124589` are two values appearing the `CUC` field of `Cust_table`, and $\langle$`124589`,`CCAAA`$\rangle$, $\langle$`CCAAA`,`-452901`$\rangle$ are two tuples appearing in the `Inv_table`. Note that, by the mapping assertions, thete two tuples satisfy the predicate `Involved` in the ontology. Now, consider the query checking whether

[3]We consider languages that are fragments of OWL 2 (Consortium 2012), the ontology web language originated from Description Logics (Baader et al. 2007).

there exists an ordinary customer who involved a special customer in a project, expressed in logic as

$\exists X \exists Y \, \texttt{Ordinary}(X), \texttt{Involved}(X, Y), \texttt{Special}(Y)$

If we simply evaluate the query by searching for the corresponding pattern in the data, we come up with the answer "false", because we cannot find any pair of elements to bind to the variables X, Y in such a way that the pattern specified by the query is satisfied in the data. However, if we consider the knowledge expressed by the ontology, then we know that, in every model of the ontology, the customer with code `CCAAA` is either ordinary, or special. For the models where `CCAAA` is ordinary, the binding $X \rightarrow \texttt{CCAAA}, Y \rightarrow \texttt{-452901}$ makes the query true, whereas for the models where `CCAAA` is special, it is the binding $X \rightarrow \texttt{124589}, Y \rightarrow \texttt{CCAAA}$ that makes the query true. It follows that the certain answer to the query is "true".

What the above example shows is that query answering in OBDM may require reasoning by cases on data (in the example, on the status of the customer `CCAAA`), and this is caused in particular by the presence of certain representation patterns in the ontology (in the example, the pattern is "every customer is either special or ordinary"). It is not difficult to see that this implies high computational complexity in the size of the data, and, unfortunately, the high cost does not seem to show up only in artificially constructed worst cases (see (Schaerf 1993)). The conclusion is that OBDM is yet another scenario where the trade-off between expressive power of the modeling language, and the complexity of reasoning is extremely relevant (Levesque and Brachman 1985).

Indeed, from the computational perspective, query answering depends on (i) the language used for the ontology, (ii) the language used to specify the queries in the mapping, and (iii) the language used for user queries. As for the first aspect, many years of research on Description Logics (Baader et al. 2003) has led to specific proposals of ontology languages suitable for OBDM. I want to briefly present one of the most successful, i.e. the one based on a family of DLs, called *DL-Lite*[4], first introduced in (Calvanese et al. 2004; 2005), which has also given rise to the OWL 2 QL profile[5] of the Web Ontology Language OWL standardized by the W3C. More specifically, I refer to *DL-Lite*$_A$, which is able to capture essentially all features of Entity-Relationship diagrams and UML Class Diagrams[6].

As usual in DLs, *DL-Lite*$_A$ allows for representing the domain of interest in terms of *concepts*, denoting sets of objects, and *roles* (or, *relations*) , denoting binary relations between objects. In *DL-Lite*$_A$, a concept is either an atomic concept C (i.e., a unary predicate) or the projection $\exists R$ or $\exists R^-$ of a role R on its first or second component, respectively. A role can be either an atomic role R or an inverse role R^-, allowing for a complete symmetry between the two directions. *DL-Lite*$_A$ includes also

Type	DL Syntax			FOL Semantics
1	C_1	$\sqsubseteq$	$[\neg]C_2$	$\forall x.C_1(x) \rightarrow [\neg]C_2(x)$
2	$\exists R^{[-]}$	$\sqsubseteq$	C	$R^{[-]}(x, y) \rightarrow C(x)$
3	C	$\sqsubseteq$	$\exists R^{[-]}$	$C(x) \rightarrow \exists y.R^{[-]}(x, y)$
4	$R_1^{[-]}$	$\sqsubseteq$	$[\neg]R_2^{[-]}$	$R_1^{[-]}(x, y) \rightarrow [\neg]R_2^{[-]}(x, y)$
5	(funct $R^{[-]}$)			$R^{[-]}(x, y) \wedge R^{[-]}(x, z) \rightarrow y = z$

Table 1: *DL-Lite*$_A$ assertions. Symbols in square brackets may or may not be present, and $R^-(x, y)$ stands for $R(y, x)$.

value attributes relating objects in classes to domain values (such as strings or integers). The *ontology*, is modeled by means of axioms that can express *inclusion* and *disjointness* between concepts or roles, and (global) *functionality* of roles (with some restrictions on the interaction between functionality and role inclusions to ensure tractability). In Table , we illustrate the conceptual modeling constructs captured by *DL-Lite*$_A$ assertions, and provide also their meaning expressed in First-Order (FO) Logic, where all variables are implicitly universally quantified. Type 1 corresponds to ISA/disjointness on concepts, type 2 to domain/range specification for a role, type 3 to mandatory participation in a role, type 4 to ISA/disjointness on roles, and type 5 to functionality assertion on a role. The DLs of the *DL-Lite* family, including *DL-Lite*$_A$, combined with specific languages for mapping specification above, have been designed so as to enjoy the *First-Order rewritability* (FO-rewritability) property: given a UCQ q and an OBDM specification $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, it is possible to compile q, $\mathcal{O}$, and $\mathcal{M}$ into a new FO query q' formulated over $\mathcal{S}$. Such query q' has the property that, when evaluated over a database D for $\mathcal{S}$, it returns exactly the certain answers for q over the OBDM system $\langle \mathcal{J}, D \rangle$, for every data source D. Each such q' is called an (FO-)*perfect rewriting* of q w.r.t. $\mathcal{J}$. Most of the proposed techniques (Calvanese et al. 2007; Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011) to achieve FO-rewritability start from a CQ or a UCQ (i.e., a set of CQs), and end up producing a UCQ that is an expansion of the initial query. They are based on variants of clausal resolution (Leitsch 1997): every rewriting step essentially corresponds to the application of clausal resolution between a CQ among the ones already generated and a concept or role inclusion axiom of the ontology. The rewriting process terminates when a fix-point is reached, i.e., no new CQ can be generated.

The results in (Calvanese et al. 2007; Poggi et al. 2008) show that, following the technique illustrated above, conjunctive query answering is indeed *first-order rewritable* in *DL-Lite*, implying that answering (unions of) conjunctive queries can be reduced to query evaluation over a relational database, for which we can rely on standard relational DBMSs. The above property also implies that CQ answering is in AC^0 (a subclass of LOGSPACE) in data complexity. Indeed, this is an immediate consequence of the fact that the complexity of the above phase of query rewriting is independent of the data source, and that the final rewritten query is an SQL expression. An important question is whether we

[4]Not to be confused with the DLs studied in (Artale et al. 2009), which form the *DL-Lite*$_{bool}$ family.

[5]`http://www.w3.org/TR/owl2-profiles/`

[6]Except for completeness of hierarchies, that is instead present in the ontology of the above example.

can further extend the ontology specification language of OBDM without losing the above nice computational property of the query rewriting phase. In (Calvanese et al. 2013) it is shown that adding any of the main concept constructors considered in Description Logics and missing in *DL-Lite*$_A$ (e.g., negation, disjunction, qualified existential restriction, range restriction) causes a jump of the data complexity of conjunctive query answering in OBDM, which goes beyond the class AC^0. This issue has been further investigated in (Artale et al. 2009). As for the query language, we note that going beyond unions of CQs is problematic from the point of view of tractability, or even decidability. For instance, adding negation to CQs causes query answering to become undecidable (Gutiérrez-Basulto et al. 2015).

This basic techniques, introduced in (Calvanese et al. 2007), has been the subject of many investigations in the last decade, with the goal of improving its performance (Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011; Kontchakov et al. 2011; Di Pinto et al. 2013; Gottlob et al. 2014a), and extending its applicability (Lenzerini, Lepore, and Poggi 2016). More generally, the issue of designing automated reasoning algorithms for query answering in OBDM has been addressed by many scientific works and projects. New ideas of how to answer queries for different ontology languages have been proposed (see, for example, (Rosati and Almatelli 2010; Chortaras, Trivela, and Stamou 2011; Gottlob et al. 2014b; Lutz and Sabellek 2017)), or various extensions to the basic ontology languages have been explored, such as extensions based on Datalog (see (Calì et al. 2010)) or on existential rules (see (Gottlob, Manna, and Pieris 2015; Grau et al. 2013; König et al. 2015)).

Finally, there has been interesting and promising work on extending query rewriting to more expressive, not necessarily first-order rewritable, ontology languages (Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011; Eiter et al. 2012; Calì, Gottlob, and Lukasiewicz 2012; Kaminski, Nenov, and Grau 2016; Bienvenu et al. 2014).

Other services

While computing certain answers of queries under the classical semantics hes been the main subject of the research investigation on OBDM, there are several other services that an OBDM system should provide. A brief overview of two services follows.

Data quality assessment. Besides ontology-mediated querying and other data management tasks, recent works argue that OBDM is a promising tool for assessing the quality of data, especially in the presence of multiple, independent data sources (Console and Lenzerini 2014; Catarci et al. 2017). Here are some of the reasons: (i) basing data quality assessments on a formal conceptualization of the domain of interest allows us to easily blur out all the meaningless details of the single data source, and focus on real data quality issues; (ii) different data sources can be analyzed using the same yardstick, i.e., the ontology, and hence accessed/compared in terms of their quality; (iii) the use of conceptualizations shared among the different assets of an organization allows for data quality assessments that are easy to present and use in many different contexts.

Quality assessment is carried out through different dimensions, such as consistency, accuracy, completeness, confidentiality etc. We briefly discuss consistency, which is the quality dimension dealing with the coherence of data. Counterexamples to consistency shows that data suffers from integrity problems, thus providing crucial information about the assets owning such data. In the literature, it is often advocated that consistency be assessed by checking whether data follow specific rules for integrity. However, in traditional approaches such rules are either implicit, or specified depending on the single data source under analysis. On the contrary, OBDM promotes a new method, where the rules to be checked are derived directly from the ontology, and have been validated by the process of building the conceptual model of the domain. In addition, instead of implementing laborious quality checking tasks for the various sources, the inference capabilities inherent in ODBM systems provide automated techniques for accessing consistency, singling out the various inconsistencies present in the data, even ranking them according to various predetermined criteria. For example, in the application scenario discussed in the introduction, we are not forced to implement a specific rule for checking whether a customer exists that is classified by the data sources both as an ordinary and as a special customer. Indeed, we can rely of the automatic verification of the rule by means of the OBDM system, as part of the consistency check of the whole OBDM system. We point out that the extensive research carried out in the last years has produced optimized algorithms for consistency checking, that scale nicely when applied to big data sources. Similar considerations hold for other data quality dimensions.

Inconsistency tolerance. What are we supposed to do once we have found out possibile consistency problems in the data sources? It is well-known that inconsistency causes severe problems in logic-based Knowledge Representation systems. Since an inconsistent logical theory has no classical model, it logically implies every formula, and therefore query answering over an inconsistent knowledge base becomes meaningless under classical logic semantics. Unfortunately, when in real world OBDM systems, inconsistencies between the domain knowledge represented by the ontology and the data at the sources are likely to occur, because data sources are generally maintained by single applications, and are not kept coherent neither with other data sources, nor with the axioms of the underlying ontology. Many research papers in the last years deals with this problem (Lembo et al. 2010; Rosati 2011; Lembo et al. 2011). In many of these approaches the fundamental tool for obtaining consistent information from an inconsistent OBDM system is the notion of repair (Arenas, Bertossi, and Chomicki 1999). A repair of a dataset contradicting a set of axioms is a database obtained by applying a "minimal" set of changes that restores consistency. There are several interpretations of the notion of "minimality", and different interpretations give rise to different inconsistency-tolerant semantics. Under most interpretations of minimality, there are many possible repairs for the system, and the approach sanctions that what is con-

sistently true is simply what is true in all possible repairs. Thus, inconsistency-tolerant query answering amounts to computing the tuples that are answers to the query in all possible repairs. Interesting papers investigating these notions in the context of OBDM are (Lembo et al. 2015; Bienvenu, Bourgaux, and Goasdoué 2016).

Open data publishing. Current practices for publishing Open Data focus essentially on providing extensional information (often in very simple forms, such as CSV files), and they carry out the task of documenting data mostly by using metadata expressed in natural languages, or in terms of record structures. As a consequence, the semantics of datasets is not formally expressed in a machine-readable form. As we said before, OBDM opens up the possibility of a new way of publishing data, with the idea of annotating data items with the ontology elements that describe them in terms of the concepts in the domain of the organization. When an OBDM specification is available in an organization, an obvious way to proceed to Open Data publication is as follows: (i) express the dataset to be published in terms of a SPARQL query over the ontology, (ii) compute the certain answers to the query, and (iii) publish the result of the certain answer computation, using the query expression and the ontology as a basis for annotating the dataset with suitable metadata expressing its semantics. Using this method, the ontology is the heart of the task: it is used for expressing the content of the dataset to be published (in terms of a query), and it is used, together with the query, for annotating the published data. First results on using OBDM for open data are reported in (Cima 2017).

Conclusions

The OBDM paradigm is relatively new, but it is attracting a strong interest from several communities. Specific tools have been designed and delivered for query answering in OBDM (see, in particular, (Calvanese et al. 2011; 2017) and `https://www.stardog.com/`), and several projects have been carried out with the goal of adopting this paradigm in real world applications (see, for example, (Kharlamov et al. 2015; Antonioli et al. 2013; Daraio et al. 2016)). From a research perspective, many groups world-wide have been working on research problems related to OBDM, producing an amazing number of scientific results (see the series of Description Logics Workshop `http://dl.kr.org/workshops/`). Interesting open problems remain, and it is reasonable to forsee that new results will contribute building novel tools or improving the current ones.

Interestingly, OBMD has helped renewing the interaction between the areas of Data Management and Artificial Intelligence. While in the last years such interaction was confined to methods and techniques for data mining and knowledge discovery, OBDM is pushing the community of Knowledge Representation and Reasoning towards research topics that are closed to Big Data and Data Science. I think that this represents a great opportunity for our community, especially in the light of the importance that the notion of data-driven society is gaining.

Acknowledgements

I would like to warmly thank Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Antonella Poggi, and Riccardo Rosati, my invaluable collaborators on the topics discussed in this paper in the last decade.

Biorgaphical statement

Maurizio Lenzerini, AAAI and ACM Fellow, is a professor of Computer Science and Engineering at the University of Rome La Sapienza. His research interests include Knowledge Representation and Reasoning, Database Theory, Ontology Languages and Reasoning about Ontologies. His current projects focus on Ontology-based Data Management, whose long-term goal is to exploit Knowledge Repreentation and Automated Reasoning techniques for addressing data access and integration issues in big data scenarios.

References

Antonioli, N.; Castanò, F.; Civili, C.; Coletta, S.; Grossi, S.; Lembo, D.; Lenzerini, M.; Poggi, A.; Savo, D. F.; and Virardi, E. 2013. Ontology-based data access: The experience at the italian department of treasury. In *Proceedings of the Industrial Track of the Conference on Advanced Information Systems Engineering 2013 (CAiSE'13), València, Spain, June 21, 2013*, 9–16.

Arenas, M.; Bertossi, L. E.; and Chomicki, J. 1999. Consistent query answers in inconsistent databases. In *Proc. of PODS'99*, 68–79.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research* 36:1–69.

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition.

Bagheri Hariri, B.; Calvanese, D.; De Giacomo, G.; Deutsch, A.; and Montali, M. 2013. Verification of relational data-centric dynamic systems with external services. In *Proc. of PODS 2013*, 163–174.

Berardi, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Mecella, M. 2003. A foundational vision of *e*-services. In *Proc. of the CAiSE 2003 Workshop on Web Services, e-Business, and the Semantic Web (WES 2003)*.

Bernstein, P. A., and Haas, L. 2008. Information integration in the enterprise. *Comm. of the ACM* 51(9):72–79.

Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. *ACM Trans. Database Syst.* 39(4):33:1–33:44.

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2016. Query-driven repairing of inconsistent dl-lite knowledge bases. In *Proceedings of the Twenty-Fifth International Joint*

Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, 957–964.

Calì, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; and Pieris, A. 2010. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proc. of LICS'10*, 228–242.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. of Web Semantics* 14:57–83.

Calvanese, D., and De Giacomo, G. 2005. Data integration: A logic-based perspective. *AI Magazine* 26(1):59–70.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Rosati, R.; and Vetere, G. 2004. *DL-Lite*: Practical reasoning for rich DLs. In *Proc. of DL 2004*, volume 104 of *CEUR, ceur-ws.org*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. *DL-Lite*: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, 602–607.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning* 39(3):385–429.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. The Mastro system for ontology-based data access. *Semantic Web Journal* 2(1):43–53.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2013. Data complexity of query answering in description logics. *Artificial Intelligence* 195:335–360.

Calvanese, D.; Cogrel, B.; Komla-Ebri, S.; Kontchakov, R.; Lanti, D.; Rezk, M.; Rodriguez-Muro, M.; and Xiao, G. 2017. Ontop: Answering SPARQL queries over relational databases. *Semantic Web* 8(3):471–487.

Catarci, T.; Scannapieco, M.; Console, M.; and Demetrescu, C. 2017. My (fair) big data. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, 2974–2979.

Chortaras, A.; Trivela, D.; and Stamou, G. B. 2011. Optimized query rewriting for OWL 2 QL. In *Proc. of CADE 2011*, 192–206.

Cima, G. 2017. Preliminary results on ontology-based open data publishing. In *Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017.*

Console, M., and Lenzerini, M. 2014. Data quality in ontology-based data access: The case of consistency. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, 1020–1026.

Consortium, W. W. W. 2012. Owl 2 web ontology language structural specification and functional-style syntax. `http://www.w3.org/TR/owl2-syntax`.

Daraio, C.; Lenzerini, M.; Leporelli, C.; Moed, H. F.; Naggar, P.; Bonaccorsi, A.; and Bartolucci, A. 2016. Data integration for research and innovation policy: an ontology-based data management approach. *Scientometrics* 106(2):857–871.

De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2018. Using ontologies for semantic data integration. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years.*, volume 31 of *Studies in Big Data*. Springer International Publishing. 187–202.

Di Pinto, F.; Lembo, D.; Lenzerini, M.; Mancini, R.; Poggi, A.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2013. Optimizing query rewriting in ontology-based data access. In *Proc. of EDBT 2013*, 561–572. ACM Press.

Doan, A.; Halevy, A. Y.; and Ives, Z. G. 2012. *Principles of Data Integration*. Morgan Kaufmann.

Eiter, T.; Ortiz, M.; Simkus, M.; Tran, T.-K.; and Xiao, G. 2012. Query rewriting for Horn-SHIQ plus rules. In *Proc. of AAAI 2012*. AAAI Press/The MIT Press.

Fan, W., and Geerts, F. 2012. *Foundations of Data Quality Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Gottlob, G.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; Schwentick, T.; and Zakharyaschev, M. 2014a. The price of query rewriting in ontology-based data access. *Artif. Intell.* 213:42–59.

Gottlob, G.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; Schwentick, T.; and Zakharyaschev, M. 2014b. The price of query rewriting in ontology-based data access. *Artificial Intelligence* 213:42–59.

Gottlob, G.; Manna, M.; and Pieris, A. 2015. Polynomial rewritings for linear existential rules. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2992–2998.

Grau, B. C.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity notions for existential rules and their application to query answering in ontologies. *J. Artif. Intell. Res.* 47:741–808.

Gutiérrez-Basulto, V.; Ibáñez-García, Y. A.; Kontchakov, R.; and Kostylev, E. V. 2015. Queries with negation and inequalities over lightweight ontologies. *J. of Web Semantics* 35:184–202.

Kaminski, M.; Nenov, Y.; and Grau, B. C. 2016. Datalog rewritability of disjunctive datalog programs and non-horn ontologies. *Artificial Intelligence* 236:90–118.

Kharlamov, E.; Hovland, D.; Jiménez-Ruiz, E.; Lanti, D.; Lie, H.; Pinkel, C.; Rezk, M.; Skjæveland, M. G.; Thorstensen, E.; Xiao, G.; Zheleznyakov, D.; and Horrocks, I. 2015. Ontology based access to exploration data at statoil. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, 93–112.

König, M.; Leclère, M.; Mugnier, M.; and Thomazo, M. 2015. Sound, complete and minimal ucq-rewriting for existential rules. *Semantic Web* 6(5):451–475.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Za-

kharyaschev, M. 2011. The combined approach to ontology-based data access. In *Proc. of IJCAI 2011*, 2656–2661.

Leitsch, A. 1997. *The Resolution Calculus*. Springer.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2010. Inconsistency-tolerant semantics for description logics. In *Proc. of RR 2010*, 103–117.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. Query rewriting for inconsistent *DL-Lite* ontologies. In *Proc. of RR 2011*.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2015. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.* 33:3–29.

Lenzerini, M.; Lepore, L.; and Poggi, A. 2016. Answering metaqueries over hi (OWL 2 QL) ontologies. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 1174–1180.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, 233–246.

Lenzerini, M. 2011. Ontology-based data management. In *Proc. of CIKM 2011*, 5–6.

Levesque, H. J., and Brachman, R. J. 1985. A fundamental tradeoff in knowledge representation and reasoning. In Brachman, R. J., and Levesque, H. J., eds., *Readings in Knowledge Representation*. Morgan Kaufmann. 41–70.

Lutz, C., and Sabellek, L. 2017. Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1181–1187.

Pérez-Urbina, H.; Horrocks, I.; and Motik, B. 2009. Efficient query answering for OWL 2. In *Proc. of ISWC 2009*, volume 5823 of *LNCS*, 489–504. Springer.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J. on Data Semantics* X:133–173.

Reiter, R. 1984. Towards a logical reconstruction of relational database theory. In Brodie, M. L.; Mylopoulos, J.; and Schmidt, J. W., eds., *On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*. Springer.

Rosati, R., and Almatelli, A. 2010. Improving query answering over *DL-Lite* ontologies. In *Proc. of KR 2010*, 290–300.

Rosati, R. 2011. On the complexity of dealing with inconsistency in description logic ontologies. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, 1057–1062.

Schaerf, A. 1993. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intelligent Information Systems* 2:265–278.

Wessels, B.; Finn, R. L.; Sveinsdottir, T.; and Wadhwa, K. 2017. *Open Data and the Knowledge Society*. Amsterdam University Press.